Report for reproducing the results of the paper “Feature Reduction using Principal Component Analysis for Opinion Mining”

by Maryana Temnyk & Stanislav Vdovych

The authors of the paper used the dataset of IMDB movie reviews gathered by Bo Pang and Lillian Lee. The dataset contained reviews and indication whether it was positive or negative. We found that dataset and used it to reproduce the results. We tried to repeat all steps described in the article using python, especially nltk library for text preprocessing and scikit-learn for model training and feature reduction. All required packages are listed in requirements.txt file.

Data preparation

Firstly we deleted all useless columns leaving only column with reviews and column with ‘pos’ or ‘neg’ tag, which were replaced by 1 and 0 respectively. After that, all reviews were tokenized, stemmed and all stop words were deleted from them. To create term-frequency matrix we used sklearn TF-IDF vectorizer, which as it turned out could also perform stemming and stop words omitting. At the end we had all the columns of that matrix as columns in the dataset, so that the data frame looked like that:

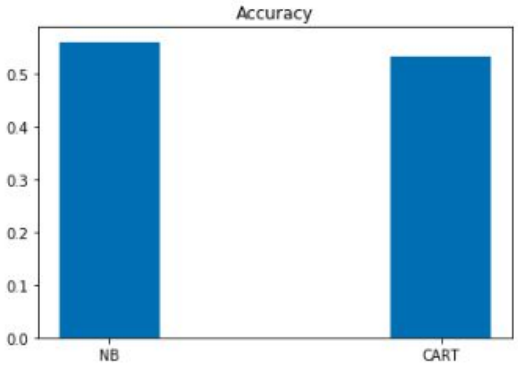
	Unnamed: 0	0	1	2	3	4	5	6	7	8	...	2569	2570	2571	2572	2573	2574	2575	2576	2577	label
0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.441014	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
3	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
4	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1

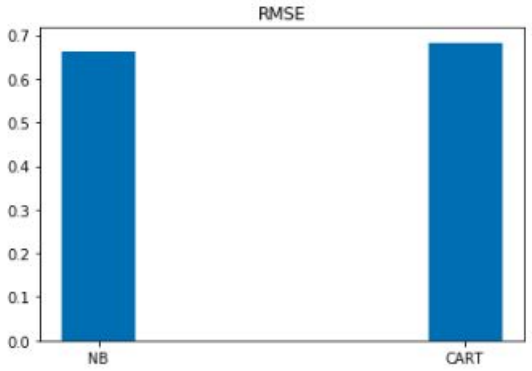
Dimentionality reduction

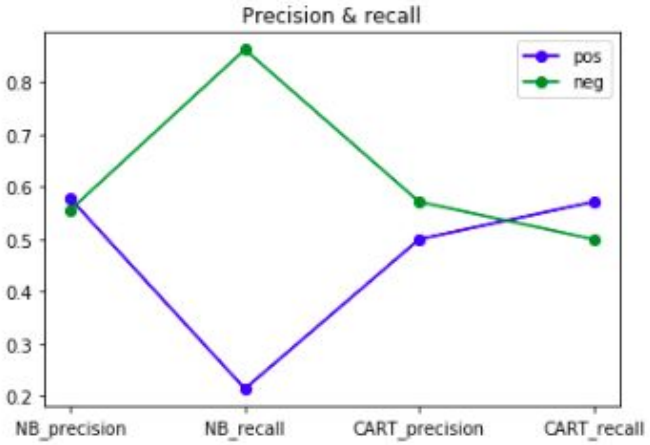
The authors decided to use principal component analysis to reduce the number of features in the dataset. Scikit-learn implementation of PCA was used to reproduce that. Firstly data was standardized using sklearn’s StandardScaler and after that we transformed our data with the PCA model. As there were no specifications in the article, we decided to leave 95% of the data variance, which left us with 286 principal components in the end, hence 286 features in now fully preprocessed dataset.

Models training

In the article authors used Learning Vector Quantization classifier, of which we had no previous knowledge, and compared its performance with CART and Naive Bayes classifiers'. Implementation of LVQ would require more time and knowledge than we currently have and there was no previous implementation of that classifier, so we did not manage to reproduce that step. but we trained CART and Naive Bayes models on our data and got such results:







Conclusions

Successes

- we reproduced most of the experiment in the paper
- we learned how to work with this type of data and how to preprocess it
- we learned about new classifiers and how they can be used

The results we received were not accurate for several reasons:

- the article was not very detailed, so we did not know some specifics of the work that researchers performed
- the lack of knowledge (LVQ case)

What could be done better:

- we could do better job with features reduction if we knew more about how to use pca in this particular case, or we could use t-sne instead
- we could tune hyperparameters in our models to get better results